

Treball de fi de Grau
Grau en Enginyeria de les Tecnologies Industrials

Disseny i implementació d'una aplicació generadora de quiz per la Formula Student

Memòria

Autor: Clara Canyís Cabutí

Director: Lluís Solano Albajes

Convocatòria: Setembre 2016



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

La Formula Student és una competició a nivell mundial on estudiants de diferents universitats dissenyen, construeixen i posen a punt un monoplaça per tal de poder competir i defençar així la feina de tot un any. Les competicions s'organitzen arreu del món i a Europa en concret n'hi ha vuit. Aquestes, degut al creixent nombre d'equips europeus, han començat a realitzar tests d'enginyeria i normativa de la competició per a decidir els equips que participaran en el seu event.

Aquest treball té com a objectiu crear una eina, en aquest cas una aplicació web, per als estudiants de l'equip ETSEIB Motorsport. Aquesta, haurà de servir per poder aprendre la normativa de la competició i preparar els qüestionaris que fan les organitzacions dels diferents events.

S'ha dissenyat, programat i validat una aplicació en html, javascript i css que genera tests amb preguntes i respostes d'una base de dades. Aquests tests poden ser generats de diferents formes com ara de aleatòriament, escollint els temes específics a practicar o simulant un test real.

Al llarg d'aquesta memòria es podrà veure una visió general del disseny del programa així com la implementació de cadascun dels elements que hi apareixen i el seu funcionament.

Finalment, cal dir que el programa creat amb aquest treball serà utilitzat per l'equip durant les pròximes temporades per tal de poder practicar i aprendre una normativa que està sempre present tant a l'hora de dissenyar qualsevol part del cotxe com a l'hora d'entrar a participar als esdeveniments i competir.

Sumari

RESUM	0
SUMARI	3
ÍNDIX DE FIGURES	5
ÍNDIX DE TAULES	7
1. GLOSSARI	9
2. PREFACI	11
2.1. Origen del projecte	11
2.2. Motivació	12
2.3. Requeriments previs	12
3. INTRODUCCIÓ	13
3.1. Objectius del projecte	13
3.2. Abast del projecte.....	13
4. ANÀLISI D'ANTECEDENTS	15
4.1. Antecedents en les competicions de la Formula Student.....	15
4.2. Antecedents dins l'equip ETSEIB Motorsport.....	15
5. ANÀLISI DEL PROBLEMA	17
5.1. Preguntes del Quiz.....	17
5.2. Respostes del Quiz	18
5.3. Dinàmica de resolució a l'equip	19
5.4. Temps de resposta	20
6. EINES UTILITZADES	21
7. PROPOSTA DE SOLUCIÓ	23
7.1. Disseny general	23
7.2. Interfície d'usuari	24
7.2.1. Funcionalitat.....	24
7.2.2. Estructura de la interfície.....	24
7.2.3. Randomly.....	26
7.2.4. Quiz Simulation.....	30
7.2.5. By topics	33
7.2.6. FSAE rules.....	34
7.3. Plantilla	35

7.3.1. Cos i text.....	35
7.3.2. NavBar	37
7.3.3. Preguntes.....	38
7.3.4. Botons.....	39
7.3.5. Rellotge.....	40
7.3.6. Signatura.....	40
7.4. Entorn Flask	41
7.4.1. Introducció	41
7.4.2. Estructura.....	41
7.4.3. Arxius Flask	42
7.5. Base de preguntes	43
8. TEST I VALIDACIÓ DEL PROGRAMA	46
8.1. Errors i resolució.....	46
8.2. Funcionament i compatibilitat	47
9. PLANIFICACIÓ	48
10. ESTUDI ECONÒMIC	49
11. IMPACTE AMBIENTAL	51
CONCLUSIONS	53
AGRAÏMENTS	55
BIBLIOGRAFIA	56
Referències bibliogràfiques	56
Bibliografia complementària	56

Índex de Figures

Figura 5.1 Gràfic de les preguntes del Quiz cada competició cada any. Font pròpia.	18
Figura 5.2 Gràfic del número de possibles respostes per pregunta. Font pròpia.....	19
Figura 5.3 Full de càlcul on es responen els questionaris de manera interna. Font pròpia. .	20
Figura 7.1: Diagrama del disseny general de l'aplicació web. Font pròpia	23
Figura 7.2 Esquema de la interfície de l'aplicació. Font pròpia.....	25
Figura 7.3: Interfície de la pàgina Randomly. Font pròpia.....	26
Figura 7.4: Codi HTML de la NavBar. Font pròpia.....	27
Figura 7.5: Codi del Selector de quantitat de preguntes. Font pròpia.....	27
Figura 7.6: Funció getRandomInt d'un interval d'enters. Font pròpia.....	28
Figura 7.7: Codi exemple d'una pregunta en HTML. Font pròpia.....	28
Figura 7.8 Interfície després de corregir el test Randomly. Font pròpia.....	29
Figura 7.9: Codi de la funció del botó START AGAIN. Font pròpia	30
Figura 7.10 Pantalla inicial de la pestanya Quiz Simulation. Font pròpia.....	30
Figura 7.11 Part del codi de correcció del Quiz Simulation. Font pròpia.....	31
Figura 7.12 Codi HTML del Timer. Font pròpia	32
Figura 7.13 Imatge del Timer en funcionament. Font pròpia	32
Figura 7.14 Codi d'inicialització del Timer. Font pròpia.....	32
Figura 7.15 Codi del timer parat quan s'acaba el quiz. Font pròpia.....	33
Figura 7.16 Pantalla inicial de la pestanya By topics. Font pròpia.	33
Figura 7.17 Codi del seleccionador de temes. Font pròpia.....	34
Figura 7.18 Codi de l'element linkat a la normativa [1]. Font pròpia.	35

Figura 7.19 Format del cos de la pàgina. Font pròpia.	36
Figura 7.20 Visualització de la propietat margin en carbassa. Font: Eines per a desenvolupadors de Google Chrome.....	36
Figura 7.21 Visualització de la propietat padding en verd. Font: Eines per a desenvolupadors de Google Chrome.....	37
Figura 7.22 Marges dels textos. Font pròpia	37
Figura 7.23 Codi css del format general de la NavBar. Font pròpia.	38
Figura 7.24 Canvi de format en passar el ratolí per sobre de About. Font pròpia.....	38
Figura 7.25 Preguntes generades per l'aplicació. Font pròpia.....	38
Figura 7.26 Codi css del format del fieldset. Font pròpia.....	39
Figura 7.27 Codi i visualització del botó actiu. Font pròpia.....	39
Figura 7.28 Codi i visualització del botó amb el cursor a sobre. Font pròpia.	39
Figura 7.29 Codi i visualització del botó inactiu. Font pròpia.	40
Figura 7.30 Codi i visualització del rellotge. Font pròpia.....	40
Figura 7.31 Codi i visualització de la signatura. Font pròpia.....	41
Figura 7.32 Estructura de carpetes i documents Flask. Font pròpia.....	42
Figura 7.33 Codi per fer córrer l'aplicació. [3]	42
Figura 7.34 Codi de inicialització de la app. [3].....	42
Figura 7.35 Codi d'associació de URL. [3].....	43
Figura 7.36 Base csv de preguntes en Excel. Font pròpia.	44
Figura 7.37 Funció que carrega les preguntes de l'arxiu csv. Font pròpia.....	44
Figura 9.1 Master plan del treball. Font pròpia.	48

Índex de taules

Taula 5.1 Nombre de preguntes per event i any. Font pròpia.	17
Taula 8.1 Errors i correccions del programa. Font pròpia.	47
Taula 10.1 Costos del projecte. Font pròpia	49
Taula 11.1 Consum i impacte ambiental del projecte. Font pròpia.	51

1. Glossari

FSAE	Formula SAE és la competició que organitza SAE Internacional i que repta als estudiants a concebre, dissenyar, fabricar i competir un monoplaça tipus formula.
FSG	Formula Student Germany és la competició de Formula Student que s'organitza cada any a Hockenheim, Alemanya.
Quiz	Test que s'ha de realitzar per a poder entrar a una competició de Formula Student i que examina bàsicament sobre aspectes de la normativa de la competició així com diversos problemes d'enginyeria.
Scrutineering	Inspecció tècnica del cotxe en el qual els jutges de la competició s'asseguren que el cotxe compleixi la normativa i sigui segur abans de poder-lo fer córrer.
HTML	HyperText Markup Language fa referència al llenguatge informàtic per a l'elaboració de pàgines web.
CSS	Cascading Style Sheets fa referència al llenguatge per a donar format i presentació a un document escrit en HTML.
NavBar	Barra de navegació on estan els hyperlinks d'una pàgina per tal de poder navegar per les pàgines d'un mateix lloc web. En aquest cas està situada horitzontalment a la part superior de la pàgina.
Timer	Relotge que compta el temps que es triga a resoldre correctament una simulació de Quiz.

2. Prefaci

2.1. Origen del projecte

L'origen del projecte té lloc gràcies a la iniciativa de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona i un grup dels seus estudiants, agrupats en l'equip *ETSEIB Motorsport*, de crear un monoplaça de competició i participar a la *Formula Student*.

La *Formula Student* és una competició universitària de caràcter internacional que té com a objectiu que estudiants universitaris realitzin les fases de disseny i construcció d'un vehicle del tipus fórmula per a poder competir a les proves que tenen lloc a diferents països. Concretament, la *Formula Student* va néixer als Estats Units l'any 1981 amb el nom de *Formula SAE* sota la coordinació de la SAE, *Society of Automotive Engineering*, i l'any 1998 per l'ImechE, *The Institution of Mechanical Engineers*, sota el nom de *Formula Student*. Actualment, se celebren competicions en altres països com per exemple Alemanya, Japó, Brasil o Espanya.

Aquests monoplaça de les diferents universitats han d'estar dissenyats i construïts tenint en compte un seguit de restriccions i normatives recollides en el document de FSAE rules [1]. Aquesta normativa dicta des de regles a seguir a l'hora de dissenyar el cotxe per tal d'assegurar la seguretat dels futurs enginyers fins a les restriccions i el funcionament que ha de tenir cadascuna de les competicions organitzades en els diferents països.

L'any 2007 va néixer l'ETSEIB Motorsport, un equip format per estudiants de l'ETSEIB per tal de participar a la *Formula Student*. Els primers integrants que van anar passant per l'equip els primers anys van dissenyar quatre cotxes de combustió - des del CAT01 fins al CAT04 - fins que, al 2011 van fer el salt per passar al cotxe elèctric. Des del 2011 fins ara s'han desenvolupat cinc cotxes elèctrics dins de l'equip - del CAT05e al CAT09e -.

Durant tots aquests anys, l'equip ETSEIB Motorsport ha participat cada any en diferents events d'aquesta competició organitzats en múltiples països: Alemanya, Regne Unit, Àustria, República Txeca, Itàlia i Espanya. En aquestes competicions, les places estan molt limitades i a causa de la tendència exponencial de creació d'equips universitaris per a participar en aquestes competicions, les organitzacions han de ser estrictes en la selecció dels equips que hi participaran.

És tal el nivell que per entrar, organitzen uns qüestionaris o Quiz amb preguntes que poden ser des de problemes d'enginyeria relacionats amb el disseny o la construcció del cotxe, com preguntes específiques de la normativa esmentada anteriorment.

2.2. Motivació

Així doncs, l'ETSEIB Motorsport es prepara durant mesos per assolir el màxim coneixement de la normativa i la màxima rapidesa a l'hora de respondre aquests Quizzes per així poder participar a les competicions de més prestigi. Fins ara, els membres de l'equip llegien i memoritzaven la normativa i es feien una o més proves internes a l'equip sobre aquesta per tal d'assegurar-se de que cadascun dels integrants la coneixia a fons.

Si bé és un sistema que no dona mals resultats, a principis d'aquesta temporada es va pensar que potser, la memorització llegint el document només al principi de la temporada, no era la forma òptima d'aprendre'l. Així és com va sorgir la idea de fer un programa interactiu que permetés aprendre la normativa i la dinàmica de resolució d'aquestes proves. A més a més, un programa d'aquestes característiques faria més atractiu a l'estudiant, d'anar revisant aquestes regles durant tota la temporada per així interioritzar-les completament.

Finalment cal dir que es podria implementar un seguiment de tal manera que tant els Team Advisors que són els professors encarregats de fer el seguiment del projecte com els Team Leaders que són els estudiants encarregats d'organitzar i gestionar l'equip poguessin estar al dia del coneixement que en tenen cadascun dels membres de l'equip.

2.3. Requeriments previs

Per tal de dur a terme aquest projecte, són necessaris coneixements bàsics de programació. Si bé el llenguatge utilitzat s'anirà aprenent i interioritzant durant el desenvolupament del projecte, sí que són necessàries nocions bàsiques de la programació i ajudarà saber el llenguatge Python.

Per últim, cal dir que per poder realitzar el projecte també és necessari conèixer el funcionament de les competicions de Formula Student així com l'estil de les preguntes dels Quiz i de la dinàmica de resolució d'aquests.

3. Introducció

El disseny i programació d'una aplicació que generi Quizzes de la Formula Student és un projecte que s'ha dut a terme durant la temporada 2015-2016 i que està pensada per poder-la implementar durant les següents temporades de l'equip ETSEIB Motorsport. En aquest projecte es descriuran els aspectes claus del seu disseny per entendre'n el funcionament.

El disseny d'aquesta aplicació, degut a la manca d'experiència dels integrants de l'equip, ha estat iteractiu. Tot i que generalment s'explicaran només els resultats finals, en els casos on es cregui adequat, s'explicarà també el procés d'implementació de les eines utilitzades i la seva evolució justificada al llarg de la durada del treball.

3.1. Objectius del projecte

L'objectiu del projecte és el de dissenyar, programar i posar en pràctica una aplicació web que permeti als membres de l'equip ETSEIB Motorsport aprendre i interioritzar la normativa de *Formula Student* recollida en el document FSAE rules [1] per tal de poder realitzar amb èxit les diferents proves d'entrada a les competicions.

En aquest treball es vol centrar l'atenció en crear una aplicació web que permeti a l'usuari interaccionar-hi i poder preparar els testos que podrien aparèixer en futurs Quizzes de la *Formula Student*. Per tal d'aconseguir això s'han determinat diferents punts que cal assolir:

- L'usuari ha de poder estudiar i practicar la normativa de FSAE [1] i la normativa específica d'Alemanya FSG [2] de diferents maneres que siguin útils per al seu aprenentatge i la posterior participació en els Quizzes de les competicions Europees.
- Tot i que el treball vol fixar-se sobretot en la part visual i interactiva de l'aplicació es farà una base de dades senzilla per tal que l'aplicació pugui funcionar correctament amb el màxim de preguntes i respostes.
- Finalment cal destacar que ha de tenir una navegació intuïtiva i que en permeti el fàcil ús.

3.2. Abast del projecte

L'abast d'aquest projecte s'emmarca a partir de la temporada 2016 – 2017 de l'equip ETSEIB Motorsport. Com que és una eina per al millor funcionament de l'equip i no un software d'un

cotxe concret, el projecte es podrà aplicar a partir de la pròxima temporada per tal de millorar les qualificacions de les proves d'entrada del gener del 2017 i es podrà anar millorant i implementant-hi altres eines que es creguin necessàries a partir d'aleshores.

Així doncs es pretén dissenyar i programar un software des de zero perquè assoleixi els objectius esmentats anteriorment. Aquest, haurà de ser el més senzill, entenedor i versàtil possible per tal de que es pugui millorar i continuar desenvolupant durant les pròximes temporades.

4. Anàlisi d'antecedents

4.1. Antecedents en les competicions de la Formula Student

Al 1981 la *Societat d'Enginyers d'Automoció* (SAE) va crear a USA la competició de la *Formula SAE*. Més tard, al 1998, la SAE i l'IMEchE, *Institut d'Enginyers Mecànics*, a Anglaterra van organitzar per primera vegada la *Formula Student*. Aquesta *Formula Student* ha organitzat des d'aleshores events en diferents països d'Europa.

La *Formula Student* d'Alemanya va ser la primera que va proposar realitzar un Quiz per poder competir en el seu event. Fins aleshores, per a poder participar, l'equip només havia d'enviar un formulari omplert amb les seves dades. Com que no hi havia gaires equips a Europa, el registre per a la competició no era cap problema. L'any 2009 per això, en pocs segons van quedar totes les places de l'event plenes. Aquest va ser el motiu pel qual l'organització de FSG va decidir canviar el tipus de registre per un on s'examinessin més els coneixements dels equips que no la rapidesa del seu internet.

L'any 2010 va tenir lloc per primera vegada el registre mitjançant un Quiz sobre la normativa de la competició [1] [2]. Durant els següents anys, altres events organitzats a Europa van començar a utilitzar aquesta forma d'escollir els equips que hi podrien competir. En algunes d'elles també van establir un Quiz per decidir l'ordre dels equips per poder passar les verificacions tècniques o Scrutineering.

4.2. Antecedents dins l'equip ETSEIB Motorsport

Fins ara l'equip ha participat cada any, com a mínim, en dues competicions europees: a Alemanya i a Espanya. A més a més també ha participat en diferents temporades a Itàlia, Regne Unit, Austria i República Txeca. Excepte l'event d'Itàlia, tots requereixen de realitzar algun Quiz, ja sigui per poder competir o per determinar l'ordre de Scrutineering.

Durant el primer quadrimestre del curs, aquests tests es preparen dins l'equip per tal de poder assolir el total d'encerts en el mínim temps possible. Per a fer això, els integrants de l'equip han d'estudiar la normativa de la competició i simular Quizzes en una sala amb tot l'equip junt, com en una prova real.

Això implica buscar dies per coincidir més de 30 persones i organitzar-se per poder resoldre uns tests que sovint no duren més de deu minuts. L'equip s'organitza per seccions i hi ha un parell de persones encarregades de respondre el test des d'un sol ordinador. Aquests fan captures de pantalla de les preguntes i les penjen al Dropbox per tal que tothom les vegi. Aleshores els membres de l'equip intenten resoldre les preguntes i van penjant les solucions

en un full de calcul del Drive. Els encarregats d'enviar les respostes marcaran les que estiguin indicades al Drive.

Com que l'organització dins de l'equip a l'hora de resoldre els testos reals ja està feta, per tal d'aconseguir baixar el temps de resposta de l'equip s'haurà d'aconseguir baixar els temps individuals. Això significa que cadascú hauria de ser capaç de resoldre les preguntes més ràpidament. Només es podrà aconseguir aquest objectiu aprenent millor la normativa i practicant preguntes que es puguin fer en les possibles proves.

5. Anàlisi del problema

Per tal de conèixer a fons el problema a resoldre s'ha fet un estudi dels tests que hi ha hagut durant els últims anys a la *Formula Student*. Estudiant aquests aspectes es podran especificar millor les funcions i el disseny de l'aplicació.

5.1. Preguntes del Quiz

Les preguntes formulades en els tests dels diferents events que es realitzen cada any a Europa canvien de forma constant. Per tal de saber com són les preguntes i quantes n'hi ha en cada Quiz s'ha fet un petit estudi. Amb aquests resultats podrem concretar el número de preguntes a mostrar en el simulador de Quizzes de l'aplicació.

	FS Germany	FS Austria	FS Spain	FS Czech Rep.
2016	12	10	10	5
2015	12	Sense dades	10	Sense dades
2014	19	12	Sense dades	5
2013	15	14	-	-
2012	12	-	-	-
2011	11	-	-	-
2010	10	-	-	-

Taula 5.1 Nombre de preguntes per event i any. Font pròpia.

Observant la taula anterior s'ha fet un gràfic per poder representar visualment el nombre de preguntes que ens trobem en cadascun dels Quiz.

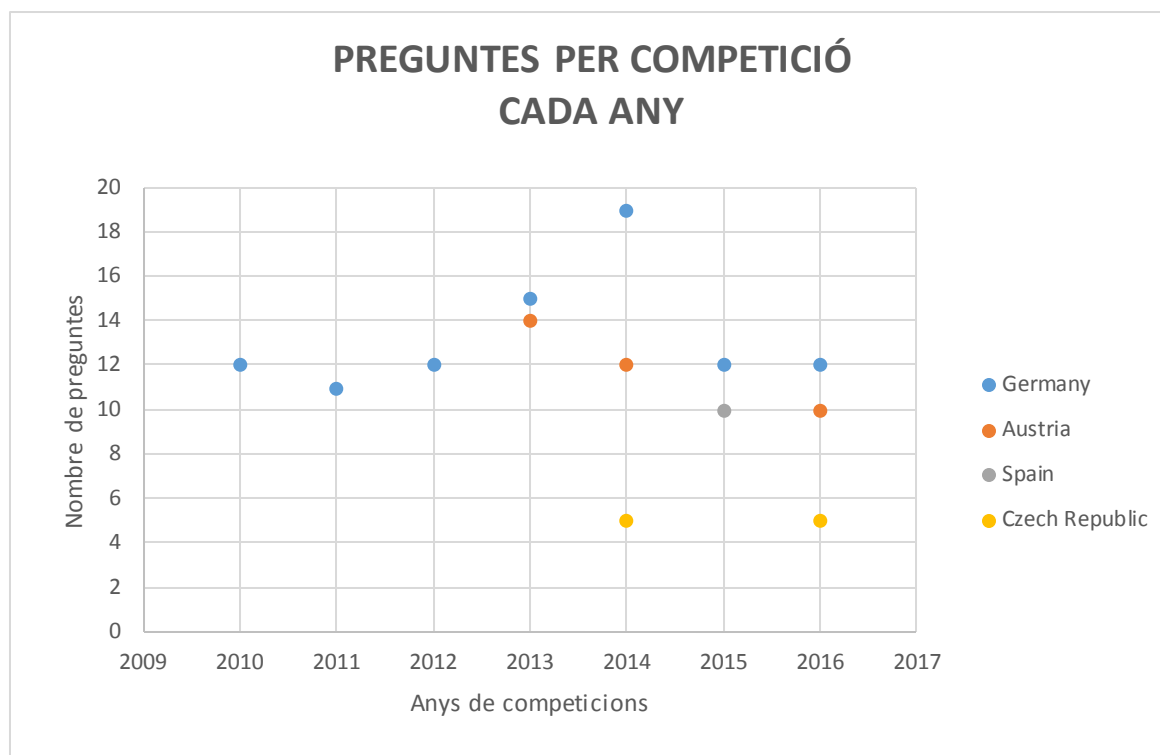


Figura 5.1 Gràfic de les preguntes del Quiz cada competició cada any. Font pròpia.

Podem veure que tots tenen un nombre de preguntes que varia entre les 5 i les 19, éssen 11 preguntes la mitjana. Tot i que no tenim gaires dades degut a el nombre limitat de testos que s'han realitzat, hem considerat que era un rang adequat.

5.2. Respostes del Quiz

Tots els testos són de tipus selecció d'una única resposta correcte. De totes maneres, la quantitat de respostes pot variar depenent de la pregunta. S'ha observat que en diferents tipus de pregunta podem trobar un nombre molt variat de possibles respostes. És per això que hem volgut estudiar la quantitat de possibles respostes per a saber com dissenyar l'aplicació.

Aquí podem veure una distribució de la quantitat de possibles respostes per a cada pregunta:

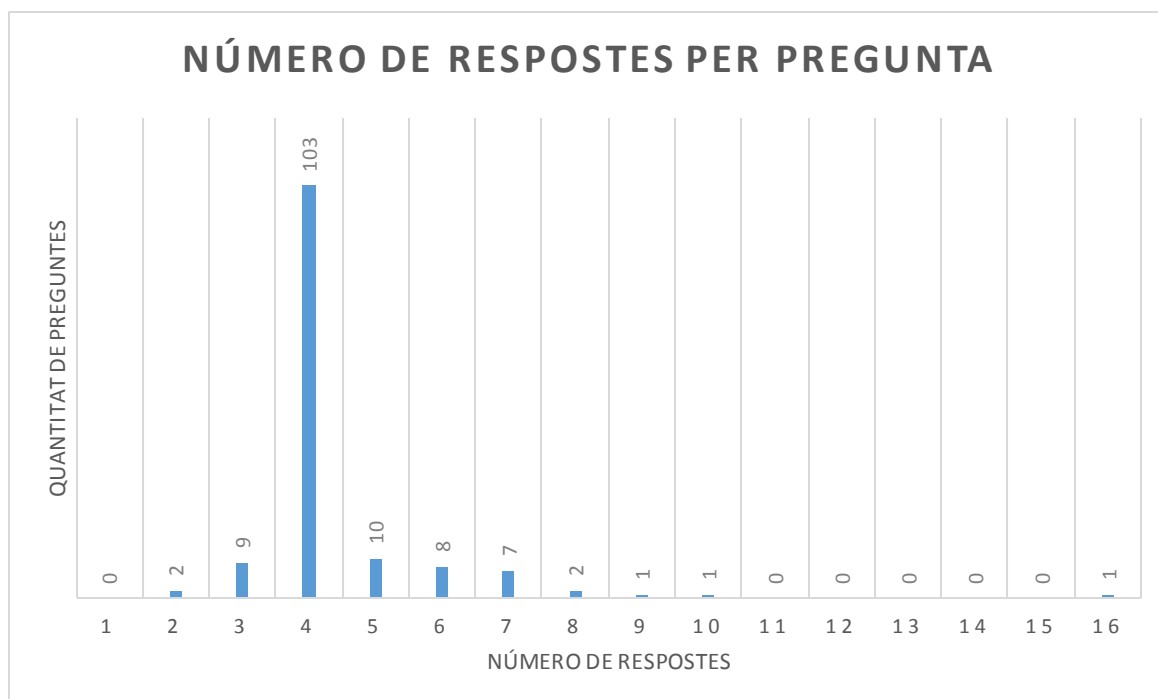


Figura 5.2 Gràfic del número de possibles respostes per pregunta. Font pròpia

Així doncs podem dir que la major part de preguntes, un 71,52%, tenen 4 possibles respostes. Per aquest motiu, el programa estarà dissenyat per tal de poder mostrar 4 possibles respostes per a cada pregunta.

5.3. Dinàmica de resolució a l'equip

A l'hora de resoldre els testos en equips, l'ETSEIB Motorsport té una dinàmica establerta per tal de optimitzar el temps de resolució d'aquest.

En primer lloc, s'obra el Quiz des d'un ordinador i es fan captures de pantalla que es pengen automàticament a una carpeta en un núvol. D'aquesta manera tots els integrants de l'equip poden veure des del seu ordinador totes les preguntes.

Aquests, es troben repartits per una sala d'informàtica i organitzats per seccions per tal de distribuir-se millor les preguntes segons el seu contingut. De manera conjunta o individual cada secció resol les qüestions que pertanyen a la seva temàtica i un cop contestades es posen a revisar les de les altres seccions.

Les respostes s'introdueixen en un full de càlcul també al núvol com el de la Figura 5.3, per a que tothom pugui veure la solució. La persona encarregada d'obrir el qüestionari també és l'encarregada de seleccionar les respostes marcades per els membres de l'equip en el qüestionari online que s'enviarà.

6. Eines utilitzades

Per tal de solucionar el problema descrit en l'apartat anterior, s'han analitzat les possibles eines a utilitzar. Com que el problema tracta d'un qüestionari web, s'ha decidit utilitzar eines web per a la creació d'una aplicació que simuli els qüestionaris que s'han de resoldre per a les competicions de la *Formula Student*.

Degut al caràcter d'aprenentatge i desenvolupament lliure del projecte, s'ha cregut adient crear l'aplicació web des de zero, fent ús de llenguatges per a programació de llocs web. Així doncs, s'ha hagut d'escollir un llenguatge per a la programació de pàgines web. D'entre tots els possibles, n'hi havia dos que podien ser adients per a la programació de l'aplicació desitjada: el Python i el HTML Dinàmic.

Python és un llenguatge que s'ha après durant el grau i que per tant, és conegut. Malgrat això, crear una aplicació web amb aquest llenguatge requereix d'eines afegides que s'haurien d'estudiar.

D'altra banda, el HTML Dinàmic està format per l'HTML, que aporta la part estàtica de l'aplicació; el Javascript, que dóna mobilitat i funcionalitat a l'HTML de l'aplicació; i el CSS que és l'encarregat de donar format o estil a tot el codi HTML visualitzat per l'usuari. Aquests tres llenguatges no són estudiats durant el grau però són bàsics a l'hora del disseny i programació de pàgines web. A més a més, donen una versatilitat enorme a l'hora de programar el que sigui necessari per ajustar-ho al disseny.

Així doncs, s'ha optat per fer servir la combinació d'HTML, JavaScript i CSS per a la programació de l'aplicació generadora de Quizzes. Amb l'HTML es farà tota la part estàtica de l'aplicació que és la part que l'usuari veu a la pantalla del seu ordinador. Exemples d'això són els botons, la barra de navegació o llistes desplegable.

D'altra banda, totes les funcions que aportin funcionalitat al programa estaran escrites amb JavaScript. Són exemple, les funcions que generin els tests o que accionin els botons de la interfície de l'usuari. Per tal de poder aconseguir unes funcions més complexes es farà ús de la llibreria de JavaScript, JQuery que permet la interacció directa entre el codi HTML i les funcions de JavaScript.

El CSS serà l'encarregat de donar format o estil al conjunt de pàgines que formen l'aplicació. Aquest llenguatge permetrà per exemple, donar la mateixa font i tamany a tots els textos o la mateixa paleta de colors a tots el quadres i botons que hi hagi a la interfície.

Per acabar, totes les preguntes i respostes de les quals es servirà el programa a l'hora de generar els qüestionaris seran extretes d'un document CSV. Es fa servir aquest format d'arxiu

perquè és fàcil d'interpretar tant amb el JavaScript a l'hora de fer-lo interactuar amb l'aplicació com per l'usuari, que mitjançant un programa com l'Excel pot afegir i modificar preguntes de manera fàcil i visual.

7. Proposta de solució

7.1. Disseny general

Per tal d'arribar als objectius determinats complint les especificacions esmentades anteriorment, s'ha proposat un disseny global com el que es mostra en la Figura 7.1.

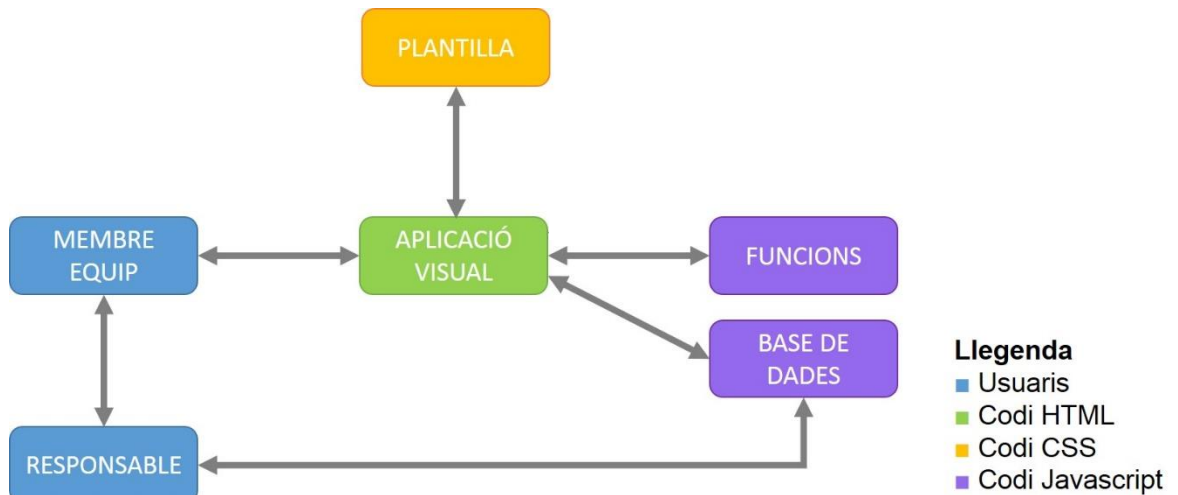


Figura 7.1: Diagrama del disseny general de l'aplicació web. Font pròpia

Com es pot apreciar en el diagrama anterior hi ha diferents blocs, cadascun d'ells programats amb llenguatges diferents. Tots ells s'han escollit pel fet que poden ser llegits i interpretats fàcilment per qualsevol navegador web, cosa que facilitarà l'ús de la web.

El sistema està pensat per tenir dos usuaris: el membre de l'equip ETSEIB Motorsport, que serà l'usuari raó de ser i el responsable de l'aplicació, que serà un usuari esporàdic. El membre de l'equip serà el que utilitzi les eines proporcionades per l'aplicació. És l'usuari que solucionarà els testos generats. El responsable en canvi, només serà l'encarregat d'actualitzar la base de preguntes que utilitzarà el sistema.

L'aplicació visual està programada en HTML i és tot allò que veu l'usuari a la pantalla del seu dispositiu electrònic. Consta de cinc pestanyes, cadascuna d'elles desenvolupada en un document .html diferent per tal de mantenir l'ordre en l'aplicació. Aquests documents estan enllaçats entre ells a través d'una barra de navegació o NavBar amb els links corresponents. En els propers apartats veurem detallades les funcions de cadascuna de les pestanyes així com els elements que contenen i el format que se'ls dona.

7.2. Interfície d'usuari

7.2.1. Funcionalitat

La interfície programada en HTML és la part de l'aplicació en la que s'ha centrat més l'atenció d'aquest treball. Necessitem una aplicació funcional que resolgui els problemes descrits en l'apartat 5 i això significa programar una interfície que funcioni correctament per al propòsit que es requereix. Se li vol donar diferents funcionalitats per tal de poder practicar els qüestionaris de diferents maneres. S'han decidit tres tipus de funcions que ha de poder servir el programa: aleatòria, per temes i simulació.

En primer lloc, l'usuari ha de ser capaç de poder generar un qüestionari amb preguntes aleatòries. Per tal de poder generar el nombre de preguntes desitjat, es disposarà un element que permeti escollir-ne el nombre exacte. Després, gràcies a un botó, es generarà el qüestionari que l'usuari podrà resoldre amb el temps que necessiti. Finalment, quan hagi acabat, podrà prémer un altre botó que corregirà el qüestionari resolt. Aquesta opció es trobarà en una de les pàgines de l'aplicació amb el nom de **Randomly**.

D'altra banda, pot ser que l'usuari vulgui practicar la normativa de la Formula Student [1] d'un únic tema, com podria ser la part elèctrica del cotxe o bé la seva estructura. En aquest cas, es donarà una altra opció perquè l'usuari pugui escollir el tema desitjat gràcies a un element de la interfície. Després de prémer el botó adient, es generarà un qüestionari amb preguntes classificades únicament amb el tema escollit. El nombre de preguntes serà d'11 degut a l'anàlisi fet en l'apartat 5.1. Aquesta pàgina de l'aplicació rebrà el nom de **By topic**.

Finalment, s'ha incorporat la possibilitat de realitzar un qüestionari amb les mateixes condicions amb les que es realitza un Quiz per entrar a un event de la competició. Així doncs, en aquest cas, es generarà un test amb un nombre aleatori de preguntes entre 5 i 19 degut a l'explicació de l'apartat 5.1. Un cop acabat el qüestionari, l'usuari haurà de prémer el botó per corregir-lo. Si tot és correcte, l'usuari haurà acabat i el temps, que haurà estat visible en tot moment a la pantalla, parerà de córrer. Si hi ha alguna resposta incorrecta, les preguntes es barrejaran, canviant l'ordre i l'usuari haurà de començar de nou, tal i com passa als Quiz reals. Aquesta pàgina de l'aplicació rebrà el nom de **Simulation Quiz**.

7.2.2. Estructura de la interfície

Abans de començar a crear el codi del programa s'ha volgut planificar bé l'estructura que tindrà aquest programa. Un cop determinades les funcions que es volen tenir, es pot estructurar tot per tal de fer-ho el més simple i entenedor possible i evitar cometre el màxim

d'errors. A continuació s'explicarà detalladament el que es pot veure a la Figura 7.2.

En primer lloc, s'ha distribuït l'aplicació en diferents pàgines. Cada pàgina està creada en un arxiu HTML diferent i estan enllaçades entre elles a través d'hyperlinks que estaran en una barra de navegació que es trobarà sempre a la part superior de la pàgina. Això s'ha fet així per tal de tenir les funcions ordenades i que es pugui passar d'una pàgina a una altra amb facilitat. Hi haurà 5 arxius HTML que estaran hyperlinkats a la barra de navegació – Home, Simulation Quiz, Randomly, By topics i About- i hi haurà un altre link que obrirà una pestanya amb la normativa de la Formula Student [apartat 8.1] [1].

Pel que fa a les funcions escrites amb JavaScript, es troben totes dins d'un mateix document amb format JS. S'han posat totes dins d'un mateix document per tal d'aconseguir reduir al màxim el codi i també per tenir de forma ordenada totes les funcions que necessita l'aplicació. D'aquesta manera només s'ha de cridar un arxiu de funcions per a totes les pàgines HTML i si es vol modificar alguna cosa és més senzill de trobar-ho. Com que les funcions que es necessiten aplicar al programa només funcionen en prémer un botó, s'ha optat per tenir una estructura de funcions que s'executen en cas de que el botó HTML corresponent hagi estat seleccionat.

Per tal de donar format CSS a tot el codi HTML de la mateixa manera, s'ha optat per fer-ho des d'un arxiu d'estil separat. D'aquesta manera, es pot donar el mateix format a totes les pàgines de l'aplicació sense haver de repetir el codi CSS per a cadascuna d'elles. Aquest arxiu, és cridat al principi de cadascuna de les pàgines, en el apartat de <head>.

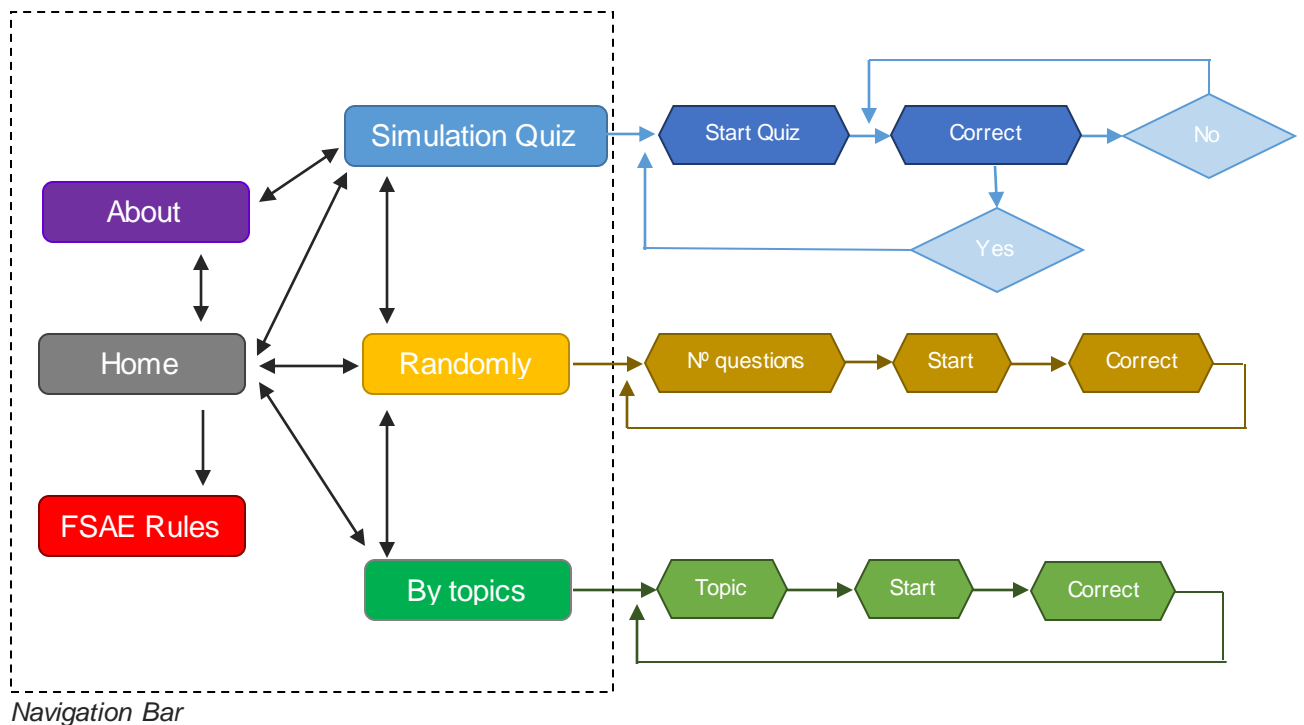


Figura 7.2 Esquema de la interfície de l'aplicació. Font pròpia.

7.2.3. Randomly

L'opció de Randomly permet a l'usuari de seleccionar la quantitat de preguntes que vol contestar i la pàgina li mostrarà aquella quantitat de preguntes amb les seves possibles respostes. El membre de l'equip haurà de respondre les preguntes i prémer el botó de CORRECT per tal que el programa li indiqui quantes preguntes ha resolt correctament.

Figura 7.3: Interfície de la pàgina Randomly. Font pròpia

La pàgina conté sis elements HTML amb les conseqüents funcions associades:

La **NavBar** que és la barra de navegació, la trobarem en totes les pàgines de la interfície del programa. Aquesta, és bàsicament un menu horitzontal desplegable que facilita la navegació entre les pàgines d'una forma completament interactiva.

Com podem veure a la Figura 7.4 es tracta d'un <nav> que és el que defineix un seguit de links de navegació. El que és dins, defineix una llista no ordenada (no numerada) d'elements i el és cada element d'aquesta llista. Com es pot veure, cada element està associat a un <a> que defineix un hyperlink que relaciona aquesta pàgina amb totes les altres.

```

<!-- Navigator Bar -->
<nav>
  <ul>
    <li><a href="../Home.html">Home</a></li>
    <li><a href="Stadistics.html">Stadistics</a></li>
    <li><a href="#" class="active">Quiz</a>
      <ul>
        <li><a href="Simulation_quiz.html">Simulation Quiz</a></li>
        <li><a href="By_topics.html">By topics</a></li>
        <li><a href="Randomly.html">Randomly</a></li>
      </ul>
    </li>
    <li><a href="About.html">About</a></li>
  </ul>
</nav>

```

Figura 7.4: Codi HTML de la NavBar. Font pròpia

El **Selector** és l'element que ens permet seleccionar quantes preguntes volem que ens mostri a la pantalla. Aquest ha restringit els valors des de 1 a 15 preguntes per tal d'habilitar un rang gran de preguntes però sense sobrepassar la quantitat que acostumen a tenir els Quiz.

```

<!-- Numero de preguntes -->
<select id='single' class="seleccionador">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
  <option>6</option>
  <option>7</option>
  <option>8</option>
  <option>9</option>
  <option>10</option>
  <option>11</option>
  <option>12</option>
  <option>13</option>
  <option>14</option>
  <option>15</option>
</select>

```

Figura 7.5: Codi del Selector de quantitat de preguntes. Font pròpia

Es tracta d'un <select>, un element de llista desplegable on cadascun dels elements de la llista és un <option>. Seleccionant una de les opcions l'element select guarda el valor de la opció seleccionada.

El **botó GO** és un element que ens permet cridar una funció un cop l'usuari prem el botó.

Aquesta, agafa el valor del selector amb el mètode de jQuery `$.val()` i amb aquest valor executa la funció que permetrà introduir un cert nombre de preguntes i respostes en la pantalla de l'usuari.

Aquesta funció crea una llista amb tants elements com el valor que s'hagi obtingut del selector i en cada element de la llista hi guarda un número aleatori, donat per la funció de la Figura 7.6, entre zero i el nombre de preguntes de la base de dades. Per generar números aleatoris fa servir la funció:

```
//Dona un numero enter aleatori entre min i max
function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min) + min);
};
```

Figura 7.6: Funció `getRandomInt` d'un interval d'enters. Font pròpia

D'aquesta manera el programa va afegint en l'element de text `<p>` del codi HTML, les preguntes i les respostes que es troben a la base de dades gràcies a la funció de JQuery `$.append()`. També aprofita per guardar les solucions d'aquestes preguntes en una llista per a la futura correcció.

En aquesta funció s'han afegit comandes després de detectar petits errors. Veieu el apartat 8.1 per a tenir informació detallada.

El codi que obtindríem d'una pregunta en el arxiu HTML seria:

```
<!-- Exemple de codi de pregunta -->
<fieldset>
    <legend>Pregunta 1 </legend>
    <p>Quan son 3*5+4-0*5?</p>
    <div id="preg1">
        <input type="radio" name="P1" id="R1" value="60" >60<br>
        <input type="radio" name="P1" id="R2" value="19" >19<br>
        <input type="radio" name="P1" id="R3" value="14" >14<br>
        <input type="radio" name="P1" id="R4" value="35" >35<br>
    </div>
</fieldset>
```

Figura 7.7: Codi exemple d'una pregunta en HTML. Font pròpia

Com podem veure a la Figura 7.7, tenim les etiquetes `<fieldset>` que són les que ens donen un espai per a cada pregunta i les seves respostes. Amb una llegenda `<legend>` es determina el número de pregunta que es tracta. Dins de l'espai d'aquesta pregunta, afegim l'enunciat amb text `<p>` i un seguit de possibles respostes `<input>`.

S'ha utilitzat inputs del tipus "radio" per tal que només es pugui seleccionar una de les respostes. El nom de la resposta fa referència al número de la pregunta, el id fa referència al

número de la resposta i el valor és el contingut de la resposta.

El **botó CORRECT** és l'encarregat de corregir les preguntes generades anteriorment amb el botó GO. Aquesta funció compara el valor de les opcions marcades amb els anteriorment guardats de la base de dades. Així doncs podrà determinar si és correcte o no i anirà sumant els punts per després mostrar-los en la secció Score.

Aquest botó també fa aparèixer el botó START AGAIN que permet a l'usuari de realitzar un altre test. A més a més, com en el cas del botó GO explicat en el apartat 8.1, aquest s'inhabilitarà quan s'hagi pitjat.

Pregunta 1

How is the size of the test resistor defined for the Insulation Monitoring Device Test?

- ☐ 250 Ohm/A
- ☐ 1000 Ohm/V
- ☒ 500 Ohm/V
- ☐ 600 V/Ohm

Pregunta 2

What does DFMA stand for in the rules?

- ☒ Design for Manufacture and Assembly
- ☐ Designated Fire Management Assistant
- ☐ Double-folded Magnetic Anchor
- ☐ Desire for Mega-Ants

Pregunta 3

Which of the following equations is a basic principle of electrical engineering?

- ☐ $R=U \cdot I$
- ☐ $R=U/I$
- ☐ $R=U+I$
- ☒ $R=U \cdot I$

Pregunta 4

Designed and coded by Clara Canyís.

Figura 7.8 Interfície després de corregir el test Randomly. Font pròpia

En aquest botó s'ha cregut necessari afegir una funcionalitat més per tal de saber quines són les preguntes correctes i quines les incorrectes. Es pot trobar l'explicació detallada a l'apartat 8.1. Així doncs quan s'ha corregit el test, es pot veure les preguntes respostes correctament i incorrectament com es veu a la Figura 7.8.

L'**element Score** és un <div> on es mostren els punts obtinguts al test. Aquest mostra la puntuació quan es prem Correct i es reseteja quan es prem el botó START AGAIN.

El **botó START AGAIN** és l'encarregat de reestablir els valors corresponents dels botons i el selector tal i com estaven al principi, abans d'utilitzar la pàgina per realitzar un test.

```

$("#button3").click(function() {
    $("#single").show();
    button1.disabled = false;
    button2.disabled = false;
    $("#formulari").empty();
    $("#puntuacio").empty();
    $("#single").val(1);
    $("#button3").hide();
});

```

Figura 7.9: Codi de la funció del botó START AGAIN. Font pròpia

En el codi de la Figura 7.9 podem veure que utilitza les funcions \$.hide() i \$.show() de JQuery per tal de mostrar i ocultar els elements necessaris i la funció \$.empty() per tal d'esborrar les preguntes i respostes que hi havia en el HTML de la pàgina. La funció \$.val() assigna al selector el valor que ha de mostrar per defecte.

7.2.4. Quiz Simulation

La pàgina de Quiz Simulation té la funció de simular un Quiz real com els que organitzen les competicions. Això significa que es tractarà d'un test d'entre 5 i 19 preguntes degut a l'explicació de l'apartat 5.1 que es definirà de forma aleatòria amb la funció de la Figura 7.5. Com s'ha dit anteriorment en aquest cas serà molt important tenir en compte el control del temps.

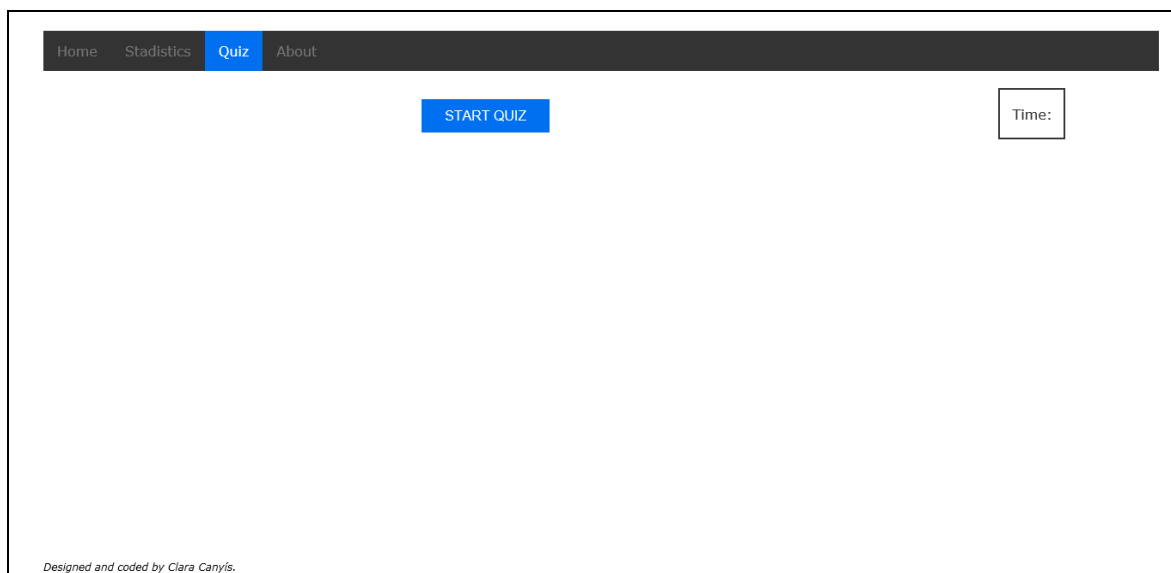


Figura 7.10 Pantalla inicial de la pestanya Quiz Simulation. Font pròpia

En aquesta pàgina hi trobem 5 elements HTML diferents:

La **NavBar** és un element que es repeteix en totes les pàgines HTML de l'aplicació web. Permet la navegació entre els diferents documents HTML de manera totalment interactiva. La seva estructura està explicada en el apartat 7.2.3.

El botó **START QUIZ** és l'encarregat de generar totes les preguntes i respostes de manera aleatòria que recull de la base de dades. Aquest botó, igual que el botó NEXT QUIZ, activa una funció que genera el Quiz.

En primer lloc, inicialitza totes les variables, esborra totes les preguntes i respostes i mostra i amaga els botons corresponents amb les funcions `$.hide()` i `$.show()`. A continuació inicialitza el timer que comença a comptar el temps.

Seguidament genera un test amb preguntes aleatòries de la base de dades i les seves respostes corresponents gràcies a la funció de JQuery `$.append()`. Finalment mostra el botó CORRECT per tal que l'usuari pugui corregir el test un cop respostes totes les preguntes.

El botó **CORRECT** es mostra un cop premut el botó START QUIZ o NEXT QUIZ. Aquest botó crida la funció encarregada de corregir tot el qüestionari comparant les respostes marcades amb la resposta correcta guardada a la base de dades.

Quan totes les respostes marcades són correctes el timer es para i es mostra un missatge per indicar que totes eren correctes. En cas que n'hi hagi una o més que no es corresponguin amb la resposta correcta, el temps continuarà corrent i les preguntes es barrejaran de manera aleatòria com passa en els Quizzes que organitzen les competicions de *la Formula Student*.

```

if (punts === numer) {
    $("#runner").runner('stop');
    $("#simulacio").empty();
    $("#simulacio").append("Great job!!");
    $("#button6").show();
    $("#button5").hide();
    time = $("#runner").text();
} else {
    //Barreja les preguntes d'ordre i torna a mostrar-ho
    $("#simulacio").empty();
    correccio = [];
    //AQUI ES BARREJA LA LLISTA
    llist_preguntes = shuffleArray(llist_preguntes);

    for (var a = 0; a < llist_preguntes.length; a++){
        $("#simulacio").append("<fieldset><legend>Pregunta " + (a + 1) + "</legend><div i
        $("#preg" + (a + 1)).append(preguntes[llist_preguntes[a]].enunciat + "<br>");
        var llista=[preguntes[llist_preguntes[a]].resposta_1,preguntes[llist_preguntes[a]
        for (var i=1; i<=4; i++){
            $("#preg" + (a + 1)).append("<input type='radio' name='P" + (a+1) + "' id='R"
        };
        correccio.push(preguntes[llist_preguntes[a]].resposta_correcte);
        $("#simulacio").append("</fieldset>");
    }

```

Figura 7.11 Part del codi de correcció del Quiz Simulation. Font pròpia.

Com veiem en el codi de la Figura 7.11, fem servir la funció *shuffleArray()* per barrejar la llista on hi ha el número de les preguntes que s'han mostrat, en cas que hi hagi alguna resposta marcada no correcta.

El **Timer** és un element `<div>` on es mostra el valor del rellotge. La seva estructura HTML és senzilla i simple ja que només consta de dos elements de text `<p>` al seu interior.

```
<!-- Timer -->
<div class="rellotge">
  <p style="display:inline-block">Time:&nbsp;</p>
  <p id="runner" style="display:inline-block"></p>
</div>
```

Figura 7.12 Codi HTML del Timer. Font pròpia

En el codi de la Figura 7.12, l'ordre *"display"* fa que es posi la paraula time i el comptador a la mateixa línia. El ` ` fa que apareixi un espai en blanc entre ambdós elements. D'aquesta forma, el timer es mostra com en la Figura 7.13.

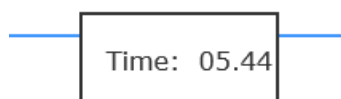


Figura 7.13 Imatge del Timer en funcionament. Font pròpia

Per tal que el Timer funcioni correctament s'ha utilitzat una llibreria de JQuery anomenada runner. Aquesta llibreria té un seguit de funcions que ens permeten fer un comptador de temps de manera senzilla i ràpida.

El primer que es fa és inicialitzar el timer amb la funció `$().runner()` quan es fa click sobre el botó START QUIZ com es mostra en la Figura 7.14.

```
//Inicialitza el timer
$('#runner').runner();
$('#runner').runner('start');
```

Figura 7.14 Codi d'inicialització del Timer. Font pròpia.

Quan es prem el botó de CORRECT de la pàgina, el timer opera segons la possibilitat de que totes les respostes estiguin bé o que n'hi hagi alguna d'incorrecte. Si totes les respostes són correctes, el runner es para com podem veure en la Figura 7.15. Si n'hi ha alguna d'incorrecte, continua comptant el temps.

```
if (punts == numer) {  
    $("#runner").runner('stop');  
}
```

Figura 7.15 Codi del timer parat quan s'acaba el quiz. Font pròpia.

El botó de **NEXT QUIZ** apareix un cop acabat un test correctament i executa la mateixa funció que el botó START QUIZ.

7.2.5. By topics

La pàgina de By topics ofereix la possibilitat de generar testos d'un sol temari. L'usuari selecciona un tema abans de fer clic al botó que genera el quiz, com es pot veure en la Figura 7.16. Aquesta opció ofereix una gran possibilitat a l'hora de preparar els quiz de les competicions ja que permet als membres de l'equip de practicar preguntes d'una sola secció. Així doncs, aquesta pàgina generarà 11 preguntes d'un tema concret.



Designed and coded by Clara Canyis.

Figura 7.16 Pantalla inicial de la pestanya By topics. Font pròpia.

En aquesta pàgina, podem trobar 5 elements HTML diferents:

La **NavBar**, que està present en totes les pàgines de l'aplicació, és la que permet la navegació entre pàgines html de manera fàcil i intuïtiva.

El **seleccionador de temes** és un <select> que conté tantes <option> com temes hi ha a la normativa [1]. Hem considerat quatre temes: Dynamics, Body, Electric i Competition. S'ha

considerat que totes les preguntes es poden classificar en aquests quatre àmbits. De totes maneres, si es volgués afegir un altre tema, només caldria afegir un altre `<option>` a continuació de l'últim element dins del `<select>` de la Figura 7.17.

```
<select id="topic" class="seleccionador">
  <option>Dynamics</option>
  <option>Electric</option>
  <option>Competition</option>
  <option>Body</option>
</select>
```

Figura 7.17 Codi del seleccionador de temes. Font pròpia

El botó **GO** és el que genera el test amb les preguntes que després apareixeran a la pantalla de l'usuari. El que fa és agafar el valor del seleccionador i, de forma aleatòria, agafa preguntes que continguin el mateix tema prèviament etiquetat a la base de preguntes. Per tal de poder agafar el valor del seleccionador s'ha d'utilitzar el mètode de jQuery `$(.text())` ja que el mètode `$(.val())` utilitzat anteriorment en la pàgina Randomly només serveix per a valors numèrics.

Un cop escollides aleatòriament, les preguntes amb les seves respostes corresponents són introduïdes en la pàgina HTML mitjançant el mètode `$.append()`.

El botó **CORRECT** és un element que ens permet corregir el qüestionari que acabem de respondre. En pitjar el botó, s'executa una funció que s'encarrega de comparar cadascun dels elements `radioButton` marcats amb els que el programa té marcats com a correctes.

En acabar, ens mostrarà un tercer botó. Aquest botó de **START AGAIN** permetrà a l'usuari de realitzar un altre test tornant a escollir el tema desitjat. Així doncs, la funció que crida aquest botó esborra totes les preguntes i respostes de la pàgina HTML i torna a mostrar el seleccionador de temes.

7.2.6. FSAE rules

Com que l'aplicació està destinada a ser una eina d'aprenentatge de la normativa [1] pels membres de l'equip ETSEIB Motorsport, s'ha cregut convenient implementar un enllaç directe al document on hi trobem totes les normes que s'han de saber i aplicar. D'una manera ràpida i senzilla l'usuari podrà consultar el pdf on hi trobarà la resposta de la major part de les preguntes.

Per tal d'implementar aquest enllaç, s'ha cregut que la `navBar` era l'element més visual i interectiu on aplicar-lo. Així doncs, s'ha afegit un darrer element `` com podem veure a la Figura 7.18.

```
<li><a href="http://www.fsaeonline.com/content/2016_FSAE_Rules.pdf">FSAE Rules 2016</a></li>
```

Figura 7.18 Codi de l'element linkat a la normativa [1]. Font pròpia.

Després d'utilitzar aquesta eina s'ha cregut necessari modificar-la per tal que la pàgina s'obri en una nova pestanya del navegador. Aquesta modificació està explicada a l'apartat 8.1.

7.3. Plantilla

Per tal de donar format a la interfície que veu l'usuari, es requereix d'una plantilla que porti la part més visual del programa. Aquesta plantilla són tot d'especificacions de format per a tots els tipus d'elements HTML que trobem a les diferents pàgines de l'aplicació.

El format d'aquest programa s'ha dissenyat en CSS per la bona compatibilitat amb l'HTML i el JavaScript i també per la seva fàcil comprensió. Totes les comandes CSS de format es troben en un mateix document per tal d'utilitzar-les en cadascuna de les pestanyes de l'aplicació per igual.

Així doncs, si es volgués canviar el format d'algun element com per exemple el color de la navBar, seria tant senzill com anar al document de la plantilla i canviar l'ordre corresponent. Immediatament totes les NavBars de les diferents pàgines de l'aplicació canviarien de color.

S'ha volgut donar un format molt senzill i visual que sigui agradable a la vista i no interfereixi amb l'utilització del programa. Per tal de no distreure l'usuari s'ha intentat fer la pàgina el màxim de simple i amb els mínims elements de format, sense línies marcades ni colors estridents. Els marges i els espaiats del text s'han intentat adequar a la lectura de l'usuari per fer-la agradable i de fàcil lectura.

7.3.1. Cos i text

Per tal de donar format al conjunt de la pàgina de l'aplicació i al text utilitzat, s'han utilitzat dos selectors amb les seves corresponents declaracions.

```
body {
  text-align: justify;
  color: black;
  margin-left: 20px;
  margin-bottom: 50px;
  padding: 20px;
  font-family: Verdana, Geneva, sans-serif;
}
```

Figura 7.19 Format del cos de la pàgina. Font pròpia.

Com podem veure a la Figura 7.19, el selector de body té sis propietats especificades. Les propietats de margin-left, margin-bottom i padding fan referència als marges i espais que trobem a l'aplicació. Per poder comprendre millor a què fa referència cadascuna de les especificacions declarades podem veure la Figura 7.20 i la Figura 7.21. La primera ens mostra l'espai que fa referència al padding de la pàgina i la segona ens mostra l'espai que fa referència al margin de la pàgina.

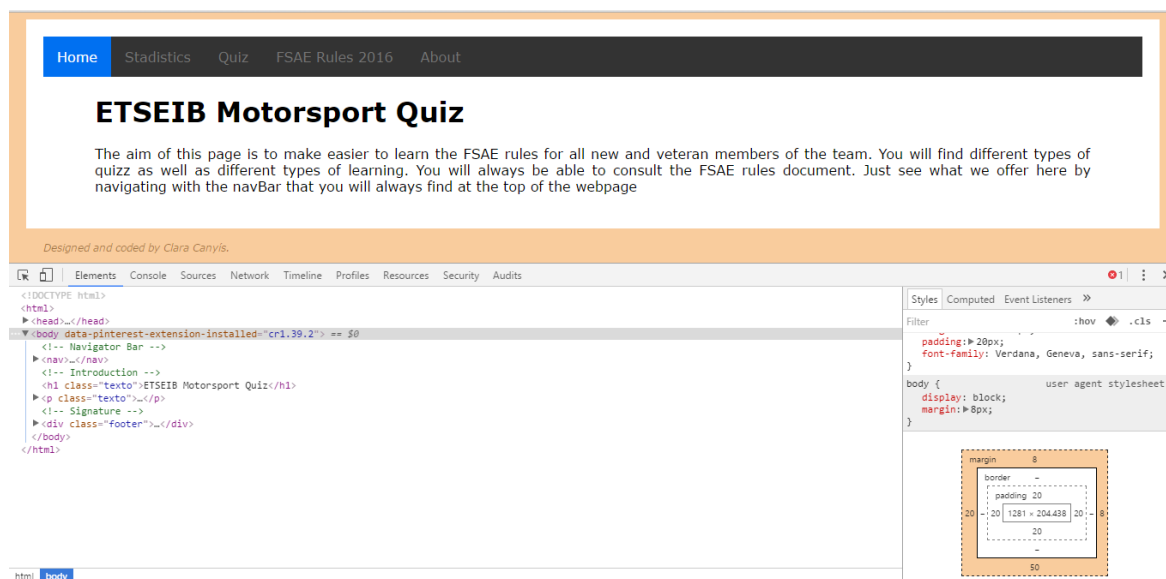


Figura 7.20 Visualització de la propietat margin en carbassa. Font: Eines per a desenvolupadors de Google Chrome.

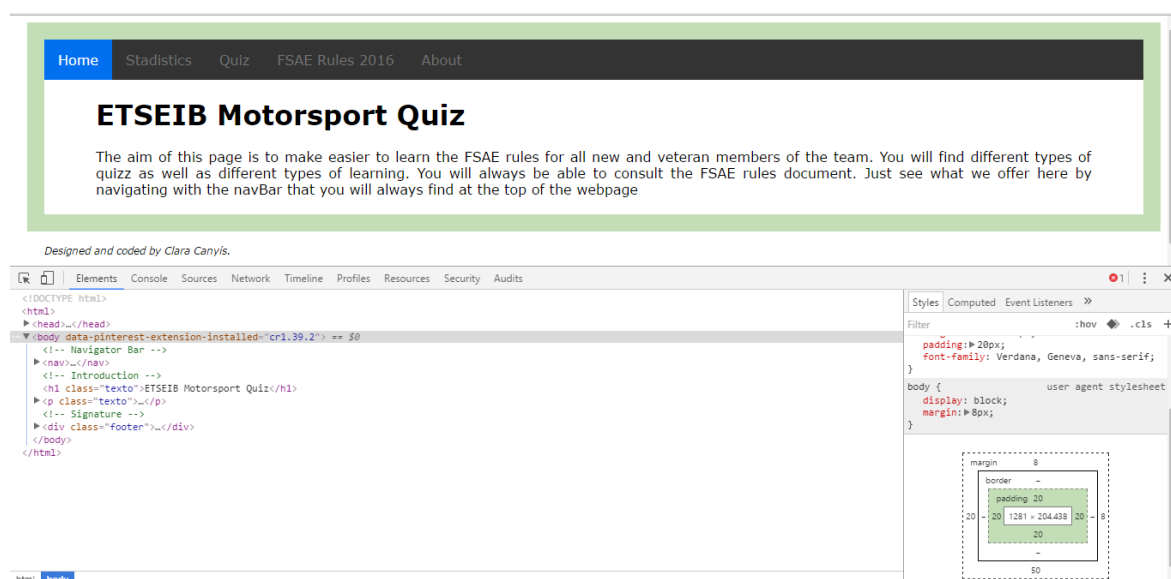


Figura 7.21 Visualització de la propietat padding en verd. Font: Eines per a desenvolupadors de Google Chrome.

Les propietats de text-align, color i font-family fan referència al text que trobarem a la pàgina. Així doncs, el text serà Verdana, justificada i negre. Els marges especificats per al format del text són els indicats en la Figura 7.22.

```
.texto {
  margin-bottom: 20px;
  margin-left: 60px;
  margin-right: 60px;
}
```

Figura 7.22 Marges dels textos. Font pròpia

7.3.2. NavBar

La barra de navegació té un sistema de format una mica més complex ja que es tracta d'un seguit de llistes de llistes. En primer lloc, tenim la barra de navegació per si sola, que té un format descrit a la Figura 7.23.

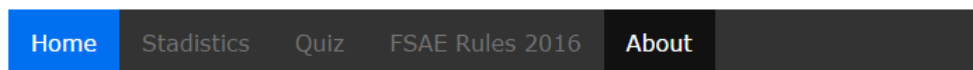
```

nav ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

```

Figura 7.23 Codi css del format general de la NavBar. Font pròpia.

Per tal que els elements canviïn de format en passar el ratolí per sobre, s'ha utilitzat el selector `:hover`. Aquest dona uns valors diferents a les propietats indicades només per quan el ratolí de l'usuari passi per sobre de l'element en concret, com podem veure a la Figura 7.24.



ETSEIB Motorsport Quiz

The aim of this page is to make easier to learn the FSAE rules for all new and as well as different types of learning. You will always be able to consult the ES

Figura 7.24 Canvi de format en passar el ratolí per sobre de About. Font pròpia.

També s'ha utilitzat un format especial amb el selector de classe `.active` pel format de la pestanya on estem actualment. En el cas de la Figura 7.24, el format de `.active` és en l'element Home.

7.3.3. Preguntes

Com podem veure a la Figura 7.25 les preguntes es troben dins d'una caixa amb la seva

Pregunta 1

All drivers have to wear underwear (long pants and long sleeve t-shirt) certified to

- ☐ SFI 3.3 or FIA 8856-2000
- ☐ SFI 3.3 or FIA 8586-2000
- ☐ SFI 2.2 or FIA 8856-2000
- ☐ SFI 33.33 or FIA 8856-2012

Pregunta 2

What is the minimum tread depth for rain tires?

- ☐ 2.4mm (3/32 inch).
- ☐ 1.6mm (2/32 inch).
- ☐ 0.8mm (1/32 inch).
- ☐ 3.2mm (4/32 inch).

Pregunta 3

Pressing one of the shut-down buttons must separate the tractive system from the accumulator block. After separating the system, the voltage in the tractive system must decrease to under 60V DC or 25V AC RMS in ♦

- ☐ less than five seconds.
- ☐ less than ten seconds

Figura 7.25 Preguntes generades per l'aplicació. Font pròpia.

pregunta i les seves possibles respostes corresponents. Per tal d'aconseguir aquest format, s'ha creat un element `<fieldset>` amb la seva `<label>` que correspon al número de pregunta.

Donant el format de `fieldset` de la Figura 7.26 s'aconsegueix la línia i l'espaiat adequats.

```
/* Per les preguntes dels Quiz */
fieldset {
    padding: 25px;
    border: 2px solid #4098FC;
    border-radius: 4px;
    margin-top: 25px;
    margin-left: 60px;
}
```

Figura 7.26 Codi css del format del `fieldset`. Font pròpia.

7.3.4. Botons

Els botons de l'aplicació tenen tots el mateix format tant quan estan actius, inactius o quan es passa el ratolí per sobre seu. A la Figura 7.27 podem veure el codi del format del botó. A la Figura 7.28 podem veure les propietats del format que varien respecte el botó actiu, quan es passa el cursor per sobre d'aquest. Finalment, a la Figura 7.29 hi ha les propietats que tornen a canviar quan el botó està inactiu— disabled.

```
/* Boto actiu */
.button {
    padding: 10px 27px;
    text-align: center;
    display: inline-block;
    font-size: 16px;
    margin-top: 15px;
    cursor: pointer;
    background-color: #0070F1;
    color: white;
    border: 2px solid white;
}
```




Figura 7.27 Codi i visualització del botó actiu. Font pròpia.

```
/* Boto amb ratolí a sobre */
.button:hover {
    background-color: white;
    color: black;
    border: 2px solid #0070F1;
}
```

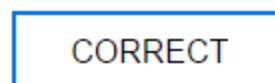


Figura 7.28 Codi i visualització del botó amb el cursor a sobre. Font pròpia.


```
/* Boto inactiu */  
.button:disabled{  
    background-color: white;  
    color: gray;  
    border: 2px solid #A9D0F5;  
}
```



Figura 7.29 Codi i visualització del botó inactiu. Font pròpia.

7.3.5. Rellotge

El rellotge és un altre element que té un format propi. En aquest cas, com es pot veure a la Figura 7.30, és un element fixat i per tant, sempre estarà a la mateixa posició de la pantalla. Aquesta és una bona opció per si hi ha moltes preguntes i es vol continuar veient el temps tot i estar al final de la pàgina.

```
/* Rellotge */  
.rellotge {  
    position: fixed;  
    right: 10%;  
    margin-top: 20px;  
    padding: 2px;  
    padding-left: 15px;  
    border: 2px solid #333;  
    background-color: white;  
    color: #333;  
}
```

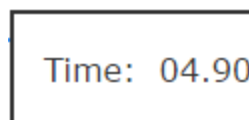


Figura 7.30 Codi i visualització del rellotge. Font pròpia.

7.3.6. Signatura

La signatura és un element que no té cap utilitat en l'aplicació. Es tracta d'un text que apareix sempre a la pàgina per marcar l'autoria de l'aplicació. Per tal que sempre aparegui a la pantalla, s'ha hagut d'utilitzar la propietat *position* amb el valor *fixed*.

Com es pot veure a la Figura 7.31 s'ha posat el fons blanc per tal que el text de la signatura no es solapés amb el de la pàgina i s'ha marcat una amplada del 100% de la finestra, per tal que no es pugui apreciar la caixa on està situat el text.

```
/* Per la signatura */  
.footer {  
    position: fixed;  
    bottom: 0px;  
    width: 100%;  
    background-color: white;  
}  
  
.signatura {  
    font-size: 12px;  
    font-style: italic;  
    margin-bottom: 10px;  
    left: 60px;  
}
```

Designed and coded by Clara Canyís.

Figura 7.31 Codi i visualització de la signatura. Font pròpia.

7.4. Entorn Flask

7.4.1. Introducció

Per tal de poder millorar l'aplicació que s'havia dissenyat i programat, es va creure necessari implementar un entorn amb Flask i així poder rebre i guardar informació necessària de manera més ràpida i senzilla.

Flask ha ajudat a poder adquirir fàcilment les preguntes i respostes de la base de dades amb qualsevol explorador d'internet. S'ha fet ús del document Flask bàsic v1.0 [3] per a poder-ho fer funcionar.

Per tal d'implementar l'aplicació a través de Flask, ha estat necessari crear una estructura de carpetes i un seguit d'arxius en Python que s'utilitzaran per inicialitzar i fer córrer el programa correctament.

7.4.2. Estructura

Per tal de dur a terme el desenvolupament a Flask, s'han estructurat els arxius del programa així com altres arxius en Python necessaris per utilitzar Flask en carpetes.

Com es pot veure a la Figura 7.32 tot el programa es troba dins de la carpeta mysite. Dins d'aquesta, hi ha les carpetes app, flask i tmp a més a més de l'arxiu run. La carpeta flask es crea automàticament i conté scripts necessaris per a poder utilitzar el Flask.

Dins la carpeta app s'hi troba l'aplicació que s'ha creat en el treball a més a més dels arxius `_init_.py` i `views.py`. Els documents que pertanyen a l'aplicació estan dividits en dues carpetes:

static, on hi ha les funcions en JavaScript, la plantilla en CSS i el document de preguntes en CSV; i templates, on hi ha els documents html de cadascuna de les pàgines de l'aplicació.

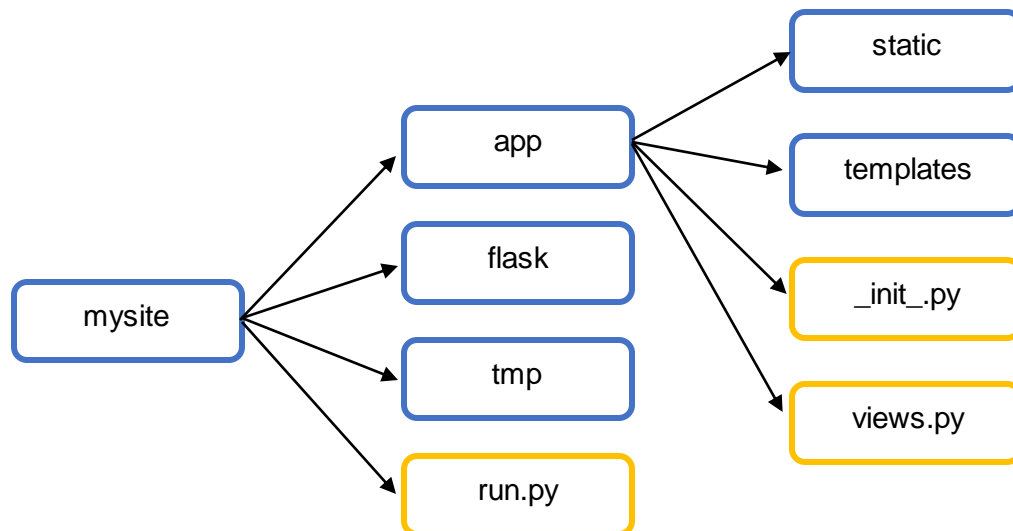


Figura 7.32 Estructura de carpetes i documents Flask. Font pròpia.

7.4.3. Arxius Flask

Per tal de fer córrer l'aplicació dins de Flask, és necessari tenir arxius de Python que s'han creat especialment per aquesta aplicació.

L'arxiu **run.py** es troba dins de la carpeta mysite. És l'arxiu que fa córrer l'aplicació web creada. Es crida des de la terminal per tal de poder utilitzar l'aplicació. Conté el codi de la Figura 7.33.

```

#!/flask/bin/python
from app import app
app.run(debug=True)
  
```

Figura 7.33 Codi per fer córrer l'aplicació. [3]

L'arxiu **_init_.py** és el responsable d'inicialitzar el programa i per tan, és el que crea l'objecte aplicació. Conté el codi de la Figura 7.34.

```

from flask import Flask

app = Flask(__name__)
from app import views
  
```

Figura 7.34 Codi de inicialització de la app. [3]

L'arxiu **views.py** és el que maneja les URLs de l'aplicació: associa una URL a cadascun dels arxius HTML que té l'aplicació guardats en la carpeta templates. Cadascuna de les direccions està indicada mitjançant una funció com es pot veure a la Figura 7.35.

```
@app.route('/')
@app.route('/Home')
def index():
    return render_template('Home.html')

@app.route('/Randomly')
def random():
    return render_template('Randomly.html')

@app.route('/By_topics')
def topics():
    return render_template('By_topics.html')

@app.route('/Simulation_quiz')
def simulation():
    return render_template('Simulation_quiz.html')

@app.route('/About')
def about():
    return render_template('About.html')
```

Figura 7.35 Codi d'associació de URL. [3]

7.5. Base de preguntes

Com ja s'ha dit en apartats anteriors, aquest treball està dedicat sobretot a la part interactiva i visual de l'aplicació així com a la seva funcionalitat. És per aquest motiu que la gestió de la base de dades de les preguntes i respostes ha quedat en un segon terme. Si bé s'ha desenvolupat per tal de poder tenir totes les preguntes i respostes, no s'ha estudiat la solució més ràpida ni eficient.

Per tal de crear la base de preguntes i respostes d'una manera ràpida i fàcil d'utilitzar s'ha optat per fer-la amb un arxiu CSV. S'ha escollit aquest format ja que és senzill d'utilitzar tant per l'aplicació com pels membres de l'equip per tal d'introduir-hi preguntes i respostes.

Per tal d'actualitzar la base de preguntes, és possible fer-ho obrint el document csv amb un programa com l'Excel. Com es pot veure a la Figura 7.36, la base de preguntes s'estructura en set columnes de les quals una és l'enunciat, quatre són les possibles respostes, una altra és la resposta correcta i l'última és l'etiqueta del tema al que pertany aquella pregunta. D'aquesta manera si es vol afegir una pregunta més a la base de dades, es pot escriure una

fila a continuació i queda immediatament inclosa dins de la base de preguntes.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Enunciat	resposta_1	resposta_2	resposta_3	resposta_4	resposta_5	Tag						
2	What is the minimum tread depth for rain tires?	2.4mm (3/32 inch)	1.6mm (2/32 inch)	0.8mm (1/32 inch)	3.2mm (4/32 inch)	2.4mm (3/32 inch)	Dynamic						
3	Which is the correct technical inspection order for E-Scrutineering, N-E-Scrutineering, E-Scrutineering, E-Scrutineering, Mechanical Competition						Competition						
4	All drivers have to wear underwear (long pants are SFI 3.3 or FIA 8856 SFI 3.3 or FIA 85 SFI 2.2 or FIA 88 SFI 33.33 or FIA SFI 3.3 or FIA 8856-2000						Competition						
5	What does DFMA stand for in the rules?	Design for Manufacture	Designated Fire	Double-folded	Desire for Mega	Design for Manufacture	Dynamic						
6	Regenerating energy is allowed and unrestricted	12 kph.	10 kph.	5 kph.	15 kph.	5 kph.	Electric						
7	Which of the following statements about the fire?	The firewall must	The firewall must	The firewall must	The firewall must	The firewall must not contain any parts	Body						
8	Which value must the current limiting resistors for 10kR	15kR	20kR	5kR	10kR		Electric						
9	Which of the following equations is a basic principle?	$R=U/I$	$R=U/I$	$R=U/I$	$R=U/I$	$R=U/I$	Electric						
10	How is the size of the test resistor defined for the 250 Ohm/A	1000 Ohm/V	500 Ohm/V	600 V/Ohm	500 Ohm/V		Electric						
11	Pressing one of the shut-down buttons must separate less than five seconds	less than five seconds	less than ten seconds	less than two seconds	more than five seconds	less than five seconds	Electric						

Figura 7.36 Base csv de preguntes en Excel. Font pròpia.

Per altra banda, el programa obra l'arxiu fent servir la funció `carregaArxiuP()` que es pot veure a la Figura 7.37. Aquesta funció utilitza la funció `$.get()` de jQuery per tal d'obrir l'arxiu csv que obra amb format text. Un cop ha aconseguit obrir-lo, s'executa un `for` per tal de transformar tot el text del csv en una llista d'objectes.

```
function carregaArxiuP() {
    var p=[];
    $.get('../static/preguntes.csv', function(data) {
        // Split the lines
        var lines = data.split('\r');
        for(var i=1; i<lines.length; i++){
            var linia = lines[i].split(';');
            p[i-1] = {enunciat: linia[0], resposta_1: linia[1], resposta_2: linia[2], resposta_3: linia[3], resposta_4: linia[4], resposta_5: linia[5], Tag: linia[6]};
        }
    }, "text");
    return p;
};
```

Figura 7.37 Funció que carrega les preguntes de l'arxiu csv. Font pròpia.

8. Test i Validació del programa

Si volem garantir el correcte funcionament del programa i evitar errors que no s'han pensat en el moment del disseny hem de testear-lo. Per tal de fer un testeig objectiu i així poder trobar més errors o coses a millorar, s'ha considerat convenient que persones de l'equip ETSEB Motorsport, que coneixen el funcionament d'aquests Quizzes, utilitzin el programa i aportin opinions sobre possibles errors o millores.

8.1. Errors i resolució

Aquests problemes o errors s'han trobat principalment deixant que persones externes al projecte provin el seu correcte funcionament. Gràcies al desconeixament del funcionament han detectat petits errors que s'han anat corregint en el programa. Podem veure'ls recollits aquí:

<p><i>Afegir preguntes quan aquestes ja s'han mostrat en pantalla.</i></p>	<p>S'ha implementat dins de la funció del botó Go una comanda que deshabilita aquest botó un cop utilitzat. Aquest es tornarà a habilitar quan ja s'hagin corregit les preguntes i l'usuari vulgui tornar a realitzar un test.</p> <p><i>Comanda utilitzada:</i></p> <pre>button1.disabled = true;</pre>
<p><i>Marcar cadascuna de les preguntes com a correctes o incorrectes.</i></p>	<p>S'ha implementat dins de la funció del botó Correct una comanda que canvia el color del marge del <fieldset> depenent de si és correcte (verd) o incorrecte (vermell).</p> <p><i>Comanda utilitzada:</i></p> <pre>\$.css("propertyname","value");</pre>
<p><i>Es pot corregir un qüestionari abans de resoldre'l.</i></p>	<p>S'ha configurat el botó perquè inicialment estigui deshabilitat i s'habiliti un cop generades les</p>

	<p>preguntes del qüestionari corresponent.</p> <p>Comanda utilitzada:</p> <pre><button disabled>CORRECT</button></pre>
<p><i>La normativa de FSAE [1] s'obra a la mateixa pestanya que l'aplicació.</i></p>	<p>Per tal de que s'obri el document en una nova pestanya s'ha afegit un atribut a l'element de la navBar de la Figura 7.18.</p> <p>Comanda utilitzada:</p> <pre>target="_blank"</pre>

Taula 8.1 Errors i correccions del programa. Font pròpia.

8.2. Funcionament i compatibilitat

Per tal de poder comprovar el correcte funcionament de l'aplicació s'ha comprovat el seu funcionament i compatibilitat amb diferents exploradors d'internet.

Amb l'ordinador s'ha comprovat el correcte funcionament i la situació adient del elements de l'aplicació amb diferents exploradors d'internet com són el Google Chrome, el Firefox i l'Internet Explorer. S'ha pogut comprovar que, fora d'algun lleuger canvi d'algun format com el de la llista desplegable o les caselles de seleccionar la resposta, no hi ha cap error. L'aplicació funciona correctament amb qualsevol dels tres programes.

9. Planificació

El projecte té una durada d'uns sis mesos: des de febrer a agost del 2016. Com tot projecte, s'ha d'establir un pla d'actuació i realització per tal de complir amb els objectius establerts en el temps necessari. En aquest cas, la durada del projecte s'ha dividit en 4 parts: aprenentatge, disseny, programació i posada a prova.

Durant l'aprenentatge s'ha hagut d'aprendre el funcionament bàsic dels llenguatges utilitzats en la programació d'aquest tipus d'aplicacions, ja que no és el llenguatge après a les assignatures d'informàtica d'aquesta escola.

En el disseny, s'han establert els objectius del projecte i com transformar-los a la realitat. S'ha pensat l'estructura i el funcionament de l'aplicació així com la possible imatge de la interfície de l'usuari per tal de fer-la ràpida i intuïtiva. També s'ha cregut necessari determinar un primer model d'emmagatzematge de les preguntes que utilitza l'aplicació.

En la fase de programació s'han anat programant els diferents mòduls especificats en el disseny i s'han fet proves i reajustos tenint en compte les limitacions del llenguatge i del temps.

Finalment per tal de poder assegurar que el programa funciona bé i que no hi ha errors no detectats durant la programació, s'han fet un seguit de proves i reajustos en l'aplicació.

Així doncs el programa de realització del projecte ha estat el següent:

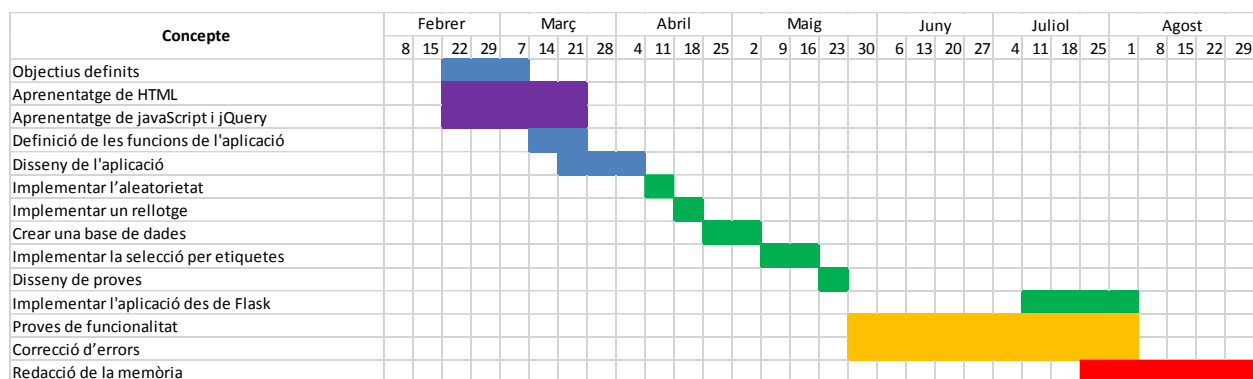


Figura 9.1 Master plan del treball. Font pròpia.

10. Estudi econòmic

Aquest projecte es basa en el disseny i la programació d'una aplicació per a la Formula Student. Per tant, en aquest apartat s'analitzaran els costos derivats de les hores de treball del personal així com del programari i altres recursos informàtics que s'hagin utilitzat.

En primer lloc, tindrem en compte els recursos humans utilitzats tant en l'anàlisi del problema per al disseny de l'aplicació com en la programació i creació final del programa. També es tindran en compte les hores de treball que ha suposat la posada a punt del programa per tal de poder-lo utilitzar des d'un servidor.

Per altra banda, s'han de comptabilitzar els equips i serveis informàtics necessaris per dur a terme el projecte, així com altre mobiliari utilitzat amb les seves pertinents amortitzacions.

Concepte	Preu	Amortització	Quantitat	Cost [€]
Recursos humans				
Anàlisi i disseny	45 €/h	-	50	2250
Desenvolupament i programació	45 €/h	-	200	9000
Posada a punt per la utilització	35 €/h	-	20	700
Redacció	25 €/h	-	30	750
Recursos informàtics				
Ordinador	700 €	10%	1	70
Llicència microsoft office	79 €	25%	1	19,75
Llicència Python	0 €	-	1	0
Llicència editor de text	0 €	-	1	0
Altres recursos materials				
Taula d'escriptori	80 €	5%	1	4
Cadira	50 €	5%	1	2,5
Electricitat	0,15 €/kwh	-	106,79	16,02
TOTAL				12 812,27 €

Taula 10.1 Costos del projecte. Font pròpia

11. Impacte ambiental

Tot i que un projecte d'aquestes característiques pot semblar que no té cap impacte ambiental aparent, s'ha de ser conscient que els recursos energètics i els equips informàtics utilitzats si que en tenen i és necessari quantificar-los.

En primer lloc, l'ordinador utilitzat per al disseny i programació de l'aplicació comporta un cert impacte ambiental tant per la seva fabricació com pel seu posterior reciclatge. Pel que fa a la fabricació, l'ordinador ja ha estat fabricat seguint diferents normatives europees i espanyoles que regulen l'impacte ambiental. És el cas, per exemple, de la Directiva 2011/65/UE que regula i restringeix l'ús de les substàncies perilloses en aparells electrònics i electrodomèstics [4].

Un cop s'acabi la vida útil de l'aparell electrònic, s'ha de fer un reciclat responsable i portar-lo en un punt Verd. Allà, seguint la Directiva 2012/19/UE [5], s'encarregaran de desmantallar l'ordinador per separar-ne els components per a la futura reutilització o bé per a reciclar el material i transformar-lo en algun altre objecte aprofitable.

Per altra banda, cal quantificar l'energia requerida durant el projecte, per tal de ser conscients de la petjada ambiental que ha produït. Per tal de convertir els kwh en kg de CO₂, grams de SO₂ i grams de NO_x s'ha utilitzat els valors donats WWF el gener del 2016 [6].

Concepte	Consum [w]	Temps [h]	Energia total [wh]	kg de CO ₂	g de SO ₂	g de NO _x
Ús de l'ordinador						
Anàlisi i disseny	350	45	15750	2,2995	4,8667	3,465
Desenvolupament i programació	350	200	70000	10,22	21,63	15,4
Posada a punt per la utilització	350	20	7000	1,022	2,163	1,54
Redacció	350	30	10500	1,533	3,2445	2,31
Il·luminació						
Disseny i desenvolupament	60	53	3180	0,4643	0,9826	0,6996
Redacció	60	6	360	0,0526	0,1112	0,0792
TOTAL				15,5914	32,998	23,4938

Taula 11.1 Consum i impacte ambiental del projecte. Font pròpia.

Conclusions

Els principals objectius d'aquest treball eren dur a terme el disseny i la programació d'una aplicació que facilités l'aprenentatge de la normativa de la Formula Student així com poder practicar els quizzs que es realitzen per poder entrar a les competicions.

S'ha aconseguit dissenyar i implementar una aplicació que genera testos amb preguntes i respostes prèviament guardades. Aquests generadors de qüestionaris funcionen correctament i han assolit l'objectiu desitjat en el sentit que és possible realitzar preguntes seleccionades a l'atzar, per temes o bé practicar els Quiz. A més a més, l'usuari és capaç de saber els errors que ha realitzat per tal de poder aprendre i millorar amb els seus errors.

Per altra banda, s'ha aconseguit proporcionar les preguntes i respostes des d'un arxiu csv extern a l'arxiu de funcions de manera que simplifica la funcionalitat. El fet de que l'arxiu sigui csv fa que sigui més senzill per l'usuari afegir i treure preguntes i respostes de la base ja que és possible fer-ho amb un programa visual i intuitiu com és l'Excel. Tots aquests aspectes confirmen l'assoliment dels objectius del treball.

A nivell personal, cal dir que aquest projecte ha permès a l'autora aprendre i familiaritzar-se amb llenguatges d'ús habitual i professional a l'hora de la programació d'aplicacions i pàgines web. L'HTML, el Javascript i el CSS són llenguatges d'ús comú que són molt útils en àmbit del món laboral.

Finalment, cal fer una autocrítica del treball i veure els aspectes que es podrien millorar o implementar de cara a futures línies de treball del projecte. La base de dades utilitzada és bàsicament el punt central de desenvolupament futur. Es pot treballar en crear una base de dades més eficient i amb més utilitat tant per adquirir i guardar preguntes com per guardar dades de l'usuari per tal de mostrar-los la seva evolució en l'aprenentatge.

Agraïments

En primer lloc, m'agraderia agrair a la direcció de l'ETSEIB per tot el suport i els recursos que dediquen cada any al projecte de Formula Student. És una excel·lent oportunitat per adquirir, a més a més de coneixaments tècnics que t'aporta el fet de crear un projecte real, coneixaments diversos com el treball en equip i l'organització i planificació.

En segon lloc, m'agraderia agrair concretament al director del projecte Lluís Solano l'ajuda i la implicació en el treball, així com el suport que ha mostrat cap a l'equip. Cal destacar les seves solucions i els exemples que ha aportat i que m'han ajudat a entendre i millorar significativament el treball. També donar les gràcies per tenir sempre alguna hora disponible per resoldre dubtes i resvisar el treball fet.

Finalment, m'agraderia agrair a tot l'equip ETSEIB Motorsport per la feina realitzada durant l'any i els bons resultats obtinguts a les competicions. En concret, donar les gràcies als companys de la secció d'electrònica que m'han ajudat i donat suport en la realització del treball. Agrair especialment al Marc Borrell per l'interès, l'ajuda i els coneixements per millorar aquest projecte.

Bibliografia

Referències bibliogràfiques

- [1] SAE INTERNATIONAL: 2016 Formula SAE® Rules.
[http://www.fsaeonline.com/content/2016_FSAE_Rules.pdf, 20 de juliol del 2016].
- [2] FSG. 2016 Formula Student Germany Rules.
[https://www.formulastudent.de/uploads/media/FSG_Rules_2016_v1.1.0_v20160314.pdf, 20 de juliol del 2016] .
- [3] LLUÍS SOLANO, *Flask bàsic v1.0*, Barcelona 2016.
- [4] DIRECTIVA 2011/65/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 8 de junio de 2011 sobre restricciones a la utilización de determinadas sustancias peligrosas en aparatos eléctricos y electrónicos.
- [5] DIRECTIVA 2012/19/UE DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 4 de julio de 2012 sobre residuos de aparatos eléctricos y electrónicos (RAEE).
- [6] Observatorio de la Electricidad Enero 2016 WWF.
[http://awsassets.wwf.es/downloads/oe_ene_2016.pdf, 10 d'agost del 2016].

Bibliografia complementària

- JQUERY. Write less, do more. [<http://api.jquery.com/>, 2016]
- W3schools. THE WORLD'S LARGEST WEB DEVELOPER SITE
[<http://www.w3schools.com/>, 2016].
- CodeAcademy. [<https://www.codecademy.com/>, 2016].
- Escola Tècnica Superior d'Enginyeria Industrial de Barcelona.
[<http://www.etseib.upc.edu/>, 2016].